Computers and Mathematics with Applications 57 (2009) 1179–1186

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

Properties of regular systems and algorithmic improvements for regular decomposition☆

Meng Jin*

LMIB, SKLSDE, Department of Mathematics, Beihang University, Beijing 100191, China

ARTICLE INFO

Article history:
Received 24 March 2008
Received in revised form 11 December 2008
Accepted 26 December 2008

Keywords:
Regular system
Constructible set
Subresultant
Algorithm
Maple

ABSTRACT

In this paper, we study the properties of regular systems and improve the efficiency of the regular decomposition method RegSer implemented in Epsilon. We define a weaker concept which retains most properties of regular system. It can be shown that from a weak regular system one can also define a regular set and vice versa. We present an algorithm RecurWeakRegSer to decompose a given polynomial system $[P, Q]$ into weak regular systems. When $Q \neq \emptyset$, the output of RecurWeakRegSer($[P, Q]$) often contains fewer components than that of RegSer($[P, Q]$). This is one advantage of RecurWeakRegSer. Another one is that RecurWeakRegSer is more efficient than RegSer. This was shown by experiments that we carried out. Since it is an essential step in RegSer to compute subresultant polynomial remainder sequences (PRS), and there is some weakness in the implementation, we implement a new version of subresultant algorithm using the optimization strategy of Ducos so that the efficiency of RegSer can be improved.

1. Introduction

The concept of regular systems was first introduced by Wang [1]. It is a generalization of the concepts of regular chains introduced by Kalkbrener [2] and proper ascending chains by Yang and Zhang [3]. Kalkbrener proposed an algorithm for decomposing a radical ideal generated by a finite set of polynomials into the intersection of unmixed ideals defined by regular chains. Several authors began to study regular chains and present various algorithms for computing them [4–6].

Wang generalized the notion of regular chains first to that of simple systems [7] and then to that of regular systems [1]. Both of them are pairs of polynomial sets and have interesting properties. The latter can be computed quite efficiently. Algorithms for computing polynomial systems into unions of regular systems or simple systems play important roles in triangular decomposition algorithms. We refer to [8] for some applications of these algorithms.

In [1], Wang showed that a regular chain is equivalent to a regular set defined by a regular system. He explored various properties of regular systems and presented an algorithm RegSer of decomposing the zeroes of a finite polynomial set or system into the union of zeroes of regular systems. Many experiments show that this method is quite efficient.

Following his work, in this paper, we try to weaken the conditions in the definition of regular systems and retain properties of that. A new algorithm called RecurWeakRegSer to decompose a given polynomial set or system into weak regular systems is presented. The motivation for our work is the weakly non-degenerate condition established in [9]. The difference is that, the weakly non-degenerate condition is used to restrict polynomials in regular sets, while it is used to restrict the "inequations" part in regular systems.

☆ This work has been supported by the Chinese National Key Basic Research (973) Projects 2004CB318000 and 2005CB321901/2 and the SKLSDE Project 07-003.

* Corresponding address: Room F427, New Main Building, Department of Mathematics, Beihang University, Beijing, China.
E-mail address: jinmeng101@gmail.com.

0898-1221/$ – see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2008.12.002

The regular decomposition method RegSer is based on computing subresultant PRS without considering any optimization. We implement a variant of subresultant algorithm using the optimization strategy of Ducos [10]. A new version NewRegSer of RegSer with this algorithm called and RecurWeakRegSer are implemented in Maple. Experiments are made to show the efficiency of the three routines.

The paper is structured as follows. Section 2 consists of preliminaries and properties of (weak) regular systems. In Section 3, first we present the algorithm RecurWeakRegSer, then discuss the three versions of algorithms of computing subresultant PRS, finally make some experiments to compare the performances of RegSer, NewRegSer and RecurWeakRegSer and algorithms mentioned above. A short conclusion is given in Section 4.

2. Notions and properties

Let k be a field of characteristic 0 and $\mathcal{K}$ be its algebraic closure. Let $x_1 \prec \cdots \prec x_n$ be n ordered variables and consider the polynomial ring $k[X] = k[x_1, \ldots, x_n]$. We refer to [7,1,11] for the definitions and notations not explicitly given here. Note that the set of the zeroes of a polynomial set $\mathbb{P}$ is denoted by $\mathbf{Z}(\mathbb{P})$.

Let $\mathbb{P}, \mathbb{Q} \subset k[X]$, and $F \in k[X]$. A *quasi-algebraic set* is the difference of the two varieties $\mathbf{Z}(\mathbb{P}) \setminus \mathbf{Z}(F)$. We denote $\mathbf{Z}(\mathbb{P}/\mathbb{Q}) = \bigcup_{Q\in\mathbb{Q}} \mathbf{Z}(\mathbb{P}) \setminus \mathbf{Z}(Q) = \mathbf{Z}(\mathbb{P}) \setminus \bigcup_{Q\in\mathbb{Q}} \mathbf{Z}(Q)$. This is called a *locally closed set*.[1] A finite union of locally closed sets is called a *constructible set* or a *quasi-variety*. See more details in [12].

We introduce some notations: $\mathbb{P}^{(i)}$ denotes $\{f \in \mathbb{P} \mid \mathrm{cls}(f) \le i\}$, $\mathbb{P}^{\langle i\rangle}$ denotes $\{f \in \mathbb{P} \mid \mathrm{cls}(f) = i\}$.

Definition 1. A polynomial system $[\mathbb{P}, \mathbb{Q}]$ is called a triangular system if $\mathbb{P}$ is a triangular set and $\mathbf{Z}(\mathbb{P}^{(i)}/\mathbb{Q}) \cap \mathbf{Z}(I) = \emptyset$ for any $I \in \mathrm{ini}(\mathbb{P}) = \{\mathrm{ini}(P) \mid P \in \mathbb{P}\}$ such that $\mathrm{cls}(I) = i$.

Definition 2. A triangular system $[\mathbb{P}, \mathbb{Q}]$ is called a regular system if the following conditions are satisfied:

(1) $\mathrm{lv}(\mathbb{P}) \cap \mathrm{lv}(\mathbb{Q}) = \emptyset$,
(2) for all $U \in \mathbb{Q}^{\langle k+1\rangle}$, and for all $(\bar{x}_1, \ldots, \bar{x}_k) \in \mathbf{Z}(\mathbb{T}^{(k)}/\mathbb{Q}^{(k)})$,

$$\mathrm{ini}(U)(\bar{x}_1, \ldots, \bar{x}_k) \neq 0, \quad k = 1, \ldots, n-1.$$

In Definition 2, there is a restriction to the initials of elements of $\mathbb{Q}$. This is not necessary and can be weakened by another condition similar to the weakly non-degenerate condition in [9].

Definition 3. A triangular system $[\mathbb{P}, \mathbb{Q}]$ is called a weak regular system if the following conditions are satisfied:

(1) $\mathrm{lv}(\mathbb{P}) \cap \mathrm{lv}(\mathbb{Q}) = \emptyset$,
(2) for all $U \in \mathbb{Q}^{\langle k+1\rangle}$, $\mathbf{Z}(\mathrm{coeff}(U)) \cap \mathbf{Z}(\mathbb{T}^{(k)}/\mathbb{Q}^{(k)}) = \emptyset$, $k = 1, \ldots, n-1$,

where $\mathrm{coeff}(U)$ is the set of all the coefficients of U viewed as a univariate polynomial in $\mathrm{lv}(U)$.

A triangular set $\mathbb{T}$ is called *regular* if there exists a polynomial set $\mathbb{U}$ such that $[\mathbb{T}, \mathbb{U}]$ is a weak regular system.

Proposition 4. *Let* $[\mathbb{T}, \mathbb{U}]$ *be a triangular system, if for all* $U \in \mathbb{U}$, $\mathbf{Z}(\mathrm{coeff}(U)) = \emptyset$, *then* $[\mathbb{T}, \mathbb{U}]$ *is a weak regular system.*

Example 1. The polynomial system $[[x^2 - 1,\ y^2 - y],\ \{(x-1)y + x - 1\}]$ with $x \prec y$ is a triangular system, but not a weak regular system since it does not satisfy the second condition in Definition 3.

Example 2. The polynomial system $[[x-1, z-y],\ \{(x-1)y - x\}]$ with $x \prec y \prec z$ is a weak regular system but not a regular system since $\mathrm{ini}((x-1)y - x) = x - 1$ vanishes at 1.

Example 3. The polynomial system $[[x - 1, w^2 - 1],\ \{zxy - 1\}]$ with $x \prec y \prec z \prec w$ is a weak regular system while the output of RegSer$([[x - 1],\ \{(y-1)z - 1\}])$ is

$$[[x-1, y, w+1], \{\}], \qquad [[x-1, w+1], \{y, zxy - 1\}],$$
$$[[x-1, w-1], \{y, zxy - 1\}], \qquad [[x-1, y, w-1], \{\}].$$

This example implies that there may exist more than necessary components when we compute a regular series of a given polynomial system.

Among various properties of a regular system $[\mathbb{T}, \mathbb{U}]$, the first nice one we think is that it is always perfect, i.e. $\mathbf{Z}(\mathbb{T}/\mathbb{U}) \neq \emptyset$. The following proposition implies that it also holds for weak regular systems.

Proposition 5. *Let* $[\mathbb{T}, \mathbb{U}]$ *be a weak regular system in* $k[X]$. *Then it is perfect in* $\mathcal{K}$.

[1] This is slightly different from the definition in [12].

Proof. Let $\mathbb{T} = [T_1, \ldots, T_r]$, we shall show by induction on r.

If $r = 0$, then $\mathbb{T} = \emptyset$ and it is obvious that $\mathbf{Z}(\emptyset/\mathbb{U}) \neq \emptyset$. Now assume the hypothesis is true for $r = k$. Let $\mathbb{T}$ be composed of $k+1$ polynomials. There are two cases needed to consider since $\mathrm{lv}(\mathbb{T}) \cap \mathrm{lv}(\mathbb{U}) = \emptyset$.

Case 1: For all $U \in \mathbb{U}$, $\mathrm{lv}(U) < \mathrm{lv}(T_{k+1})$. According to Definition 1, for all $\xi \in \mathbf{Z}(\mathbb{T}^{(k)}/\mathbb{U})$, we have $\mathrm{ini}(T_{k+1})(\xi) \neq 0$. Therefore, $T_{k+1}(\xi, x_{k+1})$ has a zero $\bar{x}$ in $\mathcal{K}$ and then $\mathbf{Z}(\mathbb{T}/\mathbb{U}) \neq \emptyset$.

Case 2: There exists $U \in \mathbb{U}$, $\mathrm{lv}(U) > \mathrm{lv}(T_{k+1})$. Divide $\mathbb{U}$ into $\mathbb{U}_1$ and $\mathbb{U}_2$ such that for all $U \in \mathbb{U}_1$, $\mathrm{cls}(U) < k+1$ and for all $U \in \mathbb{U}_2$, $\mathrm{cls}(U) > k+1$. From Case 1 we have $\mathbf{Z}(\mathbb{T}/\mathbb{U}_1) \neq \emptyset$. Assume U is the lowest one in $\mathbb{U}_2$ with respect to $\prec$ and $\mathrm{cls}(U) = j$. For any $\xi \in \mathbf{Z}(\mathbb{T}/\mathbb{U}_1)$, choose any $\bar{x}_{k+2}, \ldots, \bar{x}_{j-1}$, condition 2 of Definition 3 implies that there exists a coefficient of U in $\mathrm{lv}(U)$ which does not vanish at $\xi' = (\xi, \bar{x}_{k+2}, \ldots, \bar{x}_{j-1})$. Therefore we can choose a $\bar{x}_j$ such that $(\xi', \bar{x}_j)$ is not a zero of U. Continuing in this way, we can prove that $\mathbf{Z}(\mathbb{T}/\mathbb{U}) \neq \emptyset$. □

Given a triangular system $[\mathbb{T}, \mathbb{U}]$, let $\mathbb{T} = [T_1, \ldots, T_s]$. Sometimes for readability we denote $k[X] = k[u_1, \ldots, u_t, y_1, \ldots, y_s]$, where $y_i = \mathrm{lv}(T_i)$, $i = 1, \ldots, s$ and $u_j \in X \setminus \mathrm{lv}(\mathbb{T})$. [2]

Definition 6. A regular zero of a triangular system $[\mathbb{T}, \mathbb{U}]$ is a zero of it of the form $(u_1, \ldots, u_t, \eta_1, \ldots, \eta_s)$ where $u_1, \ldots, u_t$ are algebraically independent and $\eta_i \in k(u_1, \ldots, u_t)$, $i = 1, \ldots, s$.

The set of the regular zeroes of $[\mathbb{T}, \mathbb{U}]$ is denoted by $\mathrm{RZ}(\mathbb{T}/\mathbb{U})$. When $[\mathbb{T}, \mathbb{U}]$ is weak regular, its regular zeroes are also called the regular zeroes of $\mathbb{T}$ and denoted by $\mathrm{RZ}(\mathbb{T})$. It is well defined, since the following proposition holds.

Proposition 7. *If* $[\mathbb{T}, \mathbb{U}]$ *and* $[\mathbb{T}, \mathbb{U}']$ *are weak regular systems, then* $\mathrm{RZ}(\mathbb{T}/\mathbb{U}) = \mathrm{RZ}(\mathbb{T}/\mathbb{U}')$.

Proof. The proof is similar to that of Proposition 5.1 in [1]. The only thing needed to note is that for any $U(x) \in \mathbb{U}$, any $\xi \in \mathrm{RZ}(\mathbb{T}/\mathbb{U})$, $U(\xi) = 0$ if and only if all the coefficients of U in $\mathrm{lv}(U)$ vanish at ξ. □

Corollary 8. *Given any weak regular system* $[\mathbb{T}, \mathbb{U}]$ *and for any* $U \in \mathbb{U}$, *we have* $\mathrm{RZ}(\mathbb{T}) \cap \mathbf{Z}(U) = \emptyset$.

The following proposition gives several equivalent definitions for regular sets.

Proposition 9. *Let* $\mathbb{T} = [T_1, \ldots, T_s]$ *be a triangular set in* $k[X]$. *Denote by* $\mathbb{T}_{i-1}$ *the set of the first* $i-1$ *polynomials in* $\mathbb{T}$. *Then the following conditions are equivalent.*

(1) $\mathbb{T}$ *is a regular set.*
(2) *Let* $k = |\mathbb{T}|$, *either* $k = 1$ *or* $\mathbb{T}_{i-1}$ *is regular and* $\mathbf{Z}(\mathrm{ini}(T_i)) \cap \mathrm{RZ}(\mathbb{T}_{i-1}) = \emptyset$, $i = 2, \ldots, k$.
(3) *Either* $k = 1$ *or* $(\mathrm{ini}(T_i), \mathbb{T}_{i-1}) \cap k[u_1, \ldots, u_t] \neq (0)$, $i = 2, \ldots, k$.
(4) *Either* $k = 1$ *or there exist* $0 \neq L_i \in k[u_1, \ldots, u_t]$, $M_i \in k[X]$ *such that* $L_i \equiv M_i \mathrm{ini}(T_i) \bmod (\mathbb{T}_{i-1})$, $i = 2, \ldots, k$.

Proof. $1 \Rightarrow 2$. If $\mathbb{T}$ is regular, it is obvious if $|\mathbb{T}| = 1$. Assume $|\mathbb{T}| > 1$, then there exists a $\mathbb{U}$ such that $[\mathbb{T}, \mathbb{U}]$ is a weak regular system. According to Corollary 8, for all $\xi \in \mathrm{RZ}(\mathbb{T})$ and $U \in \mathbb{U}$, $U(\xi) \neq 0$. Therefore, $\mathrm{ini}(T)(\xi) \neq 0$ since $[\mathbb{T}, \mathbb{U}]$ is a triangular system.

$2 \Rightarrow 3$. See Theorem 1.2.1 in [13].

$3 \Rightarrow 4$. Obvious.

$4 \Rightarrow 1$. Let $\mathbb{U} = \{\mathrm{ini}(T_1), L_2, \ldots, L_k\}$. It is easy to verify by Definition 3 that $[\mathbb{T}, \mathbb{U}]$ is a weak regular system. □

In [6], Hubert defined a triangular set $\mathbb{T} = [T_1, \ldots, T_r]$ to be a regular chain if for all $k = 2, \ldots, r$, $\mathrm{ini}(T_k)$ is not a zero divisor modulo $\mathrm{sat}(\mathbb{T}^{(k-1)})$. We will show that for a given weak regular system $[\mathbb{T}, \mathbb{U}]$, polynomials in $\mathbb{U}$ also possess such property.

Proposition 10. *Let* $[\mathbb{T}, \mathbb{U}]$ *be a weak regular system in* $k[X]$. *For any* $U \in \mathbb{U}$, U *is regular with respect to* $\mathbb{T}$, *in other words,* U *is not a zero divisor modulo* $\mathrm{sat}(\mathbb{T})$.

Proof. First let us prove that $(U, \mathbb{T}) \cap k[u_1, \ldots, u_t] \neq (0)$. If not, then there exists an associated prime ideal $\mathcal{P} \in k[X]$ of $(U, \mathbb{T})$ such that $\mathcal{P} \cap k[u_1, \ldots, u_t] = (0)$. Given a generic zero $\xi = (\bar{u}_1, \ldots, \bar{u}_t, \bar{x}_1, \ldots, \bar{x}_s)$ of $\mathcal{P}$, we have $U(\xi) = 0$. Since $\mathcal{P} \cap k[u_1, \ldots, u_t] = (0)$, $\bar{u}_1, \ldots, \bar{u}_t$ are independent over k and therefore ξ is a regular zero of $\mathbb{T}$. According to Corollary 8, $U(\xi) \neq 0$, a contradiction.

Now according to Theorem 3.16 in [14], U is invertible with respect to $\mathbb{T}$. $\mathbb{T}$ is a regular set, then its elements have invertible initials. Therefore, U is regular with respect to $\mathbb{T}$. □

Proposition 11. *Let* $[\mathbb{T}, \mathbb{U}]$ *be a weak regular system in* $k[X]$. *Then*

$$\mathrm{sat}(\mathbb{T}) : \mathbb{U}^{\infty} = (\mathbb{T})$$

in $k(u_1, \ldots, u_t)[x_1, \ldots, x_s]$.

[2] Here $\mathrm{lv}(\mathbb{T}) = \{\mathrm{lv}(T) \mid T \in \mathbb{T}\}$.

Proof. According to Proposition 10, for any $U \in \mathbb{U}$, it is not a zero divisor modulo sat($\mathbb{T}$). Let $W = \Pi_{U\in\mathbb{U}}U$, W is not a zero divisor modulo sat($\mathbb{T}$) either. Therefore, sat($\mathbb{T}$) : W^∞ = sat($\mathbb{T}$). On the other hand, [$\mathbb{T}$, $\mathbb{U}$] is weak regular implies that $\mathbb{T}$ is regular, by Proposition 5.18 in [6], sat($\mathbb{T}$) is equal to the ideal ($\mathbb{T}$) in $k(u_1, \ldots, u_t)[x_1, \ldots, x_s]$. Since sat($\mathbb{T}$) : W^∞ = sat($\mathbb{T}$) : $\mathbb{U}^\infty$, we have sat($\mathbb{T}$) : $\mathbb{U}^\infty$ = ($\mathbb{T}$). □

In [1] Wang showed that for any polynomial system [$\mathbb{P}$, $\mathbb{Q}$], one can compute a finite set of regular systems [$\mathbb{T}_i$, $\mathbb{U}_i$] (called a *regular series* of [$\mathbb{P}$, $\mathbb{Q}$]) such that

$$\mathbf{Z}(\mathbb{P}/\mathbb{Q}) = \bigcup_{i=1}^{m} \mathbf{Z}(\mathbb{T}_i/\mathbb{U}_i) = \bigcup_{i=1}^{m} \mathbf{Z}(\mathrm{sat}(\mathbb{T}_i)/\mathbb{Q}). \tag{1}$$

The equation also holds for a weak regular series. Here is some property of a weak regular series.

Proposition 12. *Let* [$\mathbb{P}$, $\mathbb{Q}$] *be a polynomial system in* $k[X]$, [$\mathbb{T}_i$, $\mathbb{U}_i$] *a weak regular series of* [$\mathbb{P}$, $\mathbb{Q}$]. *Then*

$$\mathbf{Z}((\mathbb{P}) : \mathbb{Q}^\infty) = \bigcup_{i=1}^{m} \mathbf{Z}(\mathrm{sat}(\mathbb{T}_i)).$$

Proof. According to Theorem 3.2.12 (c) in [11], we have

$$\mathbf{Z}((\mathbb{P})/\mathbb{Q}) = \bigcup_{i=1}^{m} \mathbf{Z}(\mathbb{T}_i/\mathrm{ini}(\mathbb{T}_i) \cup \mathbb{Q}).$$

We claim that $\mathbf{Z}(\mathbb{T}_i/\mathrm{ini}(\mathbb{T}_i) \cup \mathbb{Q}) = \mathbf{Z}(\mathbb{T}_i/\mathrm{ini}(\mathbb{T}_i))$. If not, then $\mathbf{Z}(\mathbb{T}_i/\mathrm{ini}(\mathbb{T}_i)) \subset \mathbf{Z}(Q)$ for any $Q \in \mathbb{Q}$. Theorem 3.2.10 in [11] implies that there exists an integer $d > 0$ such that $\mathrm{prem}(Q^d, \mathbb{T}_i) = 0$, which is a contradiction with Theorem 3.2.12 (b) in [11]. Therefore, the equation $\mathbf{Z}((\mathbb{P}) : \mathbb{Q}^\infty) = \cup_{i=1}^{m} \mathbf{Z}(\mathrm{sat}(\mathbb{T}_i))$ holds. □

Proposition 13. *Let* $\mathbb{T}_1$, $\mathbb{T}_2$ *be regular sets in* $\mathcal{K}[X]$. *Then*

$$\mathrm{RZ}(\mathbb{T}_1) = \mathrm{RZ}(\mathbb{T}_2) \Leftrightarrow \sqrt{(\mathrm{sat}(\mathbb{T}_1))} = \sqrt{(\mathrm{sat}(\mathbb{T}_2))} \wedge \mathrm{lv}(\mathbb{T}_1) = \mathrm{lv}(\mathbb{T}_2)$$

in $k(u_1, \ldots, u_t)[x_1, \ldots, x_s]$.

Proof. "⇒". It is obvious that $\mathrm{lv}(\mathbb{T}_1) = \mathrm{lv}(\mathbb{T}_2)$. We will prove $\sqrt{(\mathrm{sat}(\mathbb{T}_2))} \subset \sqrt{(\mathrm{sat}(\mathbb{T}_1))}$.

For any $P \in \mathbb{T}_2$, $\mathrm{RZ}(\mathbb{T}_1) = \mathrm{RZ}(\mathbb{T}_2) \subset \mathbf{Z}(\mathbb{T}_2)$ implies that $\mathbf{Z}(\mathrm{sat}(\mathbb{T}_1)) \subset \mathbf{Z}(P)$ (Proposition 3. 2. 9 (b) in [11]). By the Hilbert Nullstellsatz, $P \in \sqrt{\mathrm{sat}(\mathbb{T}_1)}$ and then $(\mathbb{T}_2) \subset \sqrt{\mathrm{sat}(\mathbb{T}_1)}$. We have $\mathrm{sat}(\mathbb{T}_2) = (\mathbb{T}_2)$ in $k(u_1, \ldots, u_t)[x_1, \ldots, x_s]$ by Proposition 5.18 in [6]. Therefore $\sqrt{\mathrm{sat}(\mathbb{T}_2)} \subset \sqrt{\mathrm{sat}(\mathbb{T}_1)}$. The other direction is similar.

"⇐". For any $P \in \mathbb{T}_2 \subset \sqrt{(\mathrm{sat}(\mathbb{T}_1))}$, we have $\mathbf{Z}(\mathrm{sat}(\mathbb{T}_1)) \subset \mathbf{Z}(P)$. Then $\mathrm{RZ}(\mathbb{T}_1) \subset \mathbf{Z}(P) \subset \mathbf{Z}(\mathbb{T}_2)$. Since $\mathrm{lv}(\mathbb{T}_1) = \mathrm{lv}(\mathbb{T}_2)$, we have $\mathrm{RZ}(\mathbb{T}_1) \subset \mathrm{RZ}(\mathbb{T}_2)$.The other direction is similar. □

3. Algorithms and experiments

3.1. Computing weak regular series

For constructible sets, there are two types of representation methods: one is using Gröbner basis; the other takes advantage of polynomial system [15]. By regular decomposition one can represent that easily in the sense of the latter.

In this section, we show how to represent a constructible set as a finite union of weak regular systems. Roughly speaking, one can obtain a variant of RegSer by modifying R2.2.4 of it. To be self-contained, we give an alternative recursive algorithm.

Proposition 14. *Given a constructible set* $\bigcup_{i=1}^{k} \mathbf{Z}(\mathbb{P}_i/\mathbb{Q}_i)$, *there exists an algorithm to represent it as a finite union of special locally closed sets defined by weak regular systems.*

We only have to prove that for any $\mathbf{Z}(\mathbb{P}/\mathbb{Q})$, there exists a finite set of weak regular systems [$\mathbb{T}_i$, $\mathbb{U}_i$] such that

$$\mathbf{Z}(\mathbb{P}/\mathbb{Q}) = \bigcup_{i=1}^{l} \mathbf{Z}(\mathbb{T}_i/\mathbb{U}_i).$$

Lemma 15. *Given a locally closed set* $\mathbf{Z}(\mathbb{P}/\mathbb{Q})$, *there exists an algorithm RecurWeakRegSer returning a finite set of weak regular systems* [$\mathbb{T}_i$, $\mathbb{U}_i$] *such that*

(1) $\mathbf{Z}(\mathbb{T}_i/\mathbb{U}_i) \neq \emptyset$,
(2) $\mathbf{Z}(\mathbb{P}/\mathbb{Q}) = \bigcup_{i=1}^{l} \mathbf{Z}(\mathbb{T}_i/\mathbb{U}_i)$.

We show the correctness and termination of this algorithm.

Correctness. Given a locally closed set $\mathbf{Z}(\mathbb{P}/\mathbb{Q})$, in other words, a polynomial system $[\mathbb{P}, \mathbb{Q}]$, and a set of ordered variables, RecurWeakRegSer calls the recursive sub-algorithm Branch. For every k from n to 1, Branch calls PartialTriangularizeT and EliminateU. PartialTriangularizeT is borrowed from RegSer. It aims at making polynomials of class k in $\mathbb{P}$ unique and output a new triplet $[T, U, k']$. Bifurcation takes place making some initials of polynomials of class k to become zero and Branch is called.

After that, EliminateU tries to make either T_k or U_k be the empty set. T_k and U_k are respectively sets of polynomials of class k in T and U. Meanwhile, Bifurcations take place for the same reason as above. If U_k is not empty, for every element of it, say p, consider the coefficients of p in x_k. If there exists some constant coefficient, then $\mathbf{Z}(\text{coeff}(U)) = \emptyset$, then no bifurcation takes place. Otherwise, choose the simplest one, say, the coefficient with minimal leading degree and minimal terms, add it to T and make a bifurcation. After a traverse of k from num to 1, Branch outputs a triplet which is easily to be seen a weak regular system. Therefore the first condition is satisfied. For every splitting in the algorithm, the second condition always holds.

Termination. The termination of RecurWeakRegSer is due to that of PartialTriangularizeT and EliminateU which involve splitting. Given a polynomial system $[\mathbb{P}, \mathbb{Q}]$, we show the termination by induction on k. For $k = 1$, in PartialTriangularizeT, the number of polynomials in T_1 decreases strictly, and no splitting would happen since all initials are constants. If either T_1 or U_1 is the empty set, then the algorithm terminates. Otherwise, EliminateU is called. If T_1 is not empty, the degree of the polynomial in T_1 will strictly decrease and Branch is called, this call is obviously finite. Else, Q_1 is not the empty set, no splitting would happen because of constant initials. Therefore, the algorithm terminates.

Assume the termination is true for $< k$. In PartialTriangularizeT, the number and degree of polynomials in T_k decrease strictly, and so splitting times is finite. For the same reason in EliminateU, combine with our hypothesis, the algorithm terminates.

Algorithm 1: RecurWeakRegSer

Input: Polynomial system $[P, Q]$, ordered variable $x_1 \prec \cdots \prec x_n$
Output: A set of weak regular systems

```
1.1 F := P;
1.2 G := Q;
1.3 num := n;
1.4 OUTSET := ∅;
1.5 # global variable which collects all weak regular systems
1.6 Branch ([F, G, num]);
```

Algorithm 2: Branch

Input: A list in the form $[F, G, num]$
Output: A weak regular series of $[F, G]$

```
2.1  k := num;
2.2  while k > 0 do
2.3  |  L := PartialTriangularizeT ([F, G, k]);
2.4  |  # output either [T, U, l] or [T, U, l, T_k, U_k]
2.5  |  if nops(L) > 3 then
2.6  |  |  L := EliminateUk (L);
2.7  |  |  k := L[3];
2.8  |  |  if k = 0 then break;
2.9  |  else if L[3] = 0 then
2.10 |  |  break;
2.11 |  end
2.12 end
2.13 if 1 ∉ L[1] then
2.14 |  OUTSET:= OUTSET ∪{[L[1], L[2]]};
2.15 end
```

3.1.0.1. Remark

Note that R2.2.4 of Algorithm RegSer is designed to guarantee that in a given polynomial system $[\mathbb{T}, \mathbb{U}]$, the initial of every polynomial in $\mathbb{U}$ does not vanish at any zero of $\mathbb{T}$. However, this is not necessary when computing a weak regular system. We only need to restrict one coefficient of a polynomial with respect to its leading variable to be non-zero. Another note is, the "if" statement in line 21 of Algorithm 4 is an application of Proposition 4. The reason is that, if some of the elements of CoefP in Algorithm 4 is a constant, then $\mathbf{Z}(\text{coeff}(P)) = \emptyset$, and then $\mathbf{Z}(\mathbb{T}/\mathbb{U}) \neq \emptyset \Leftrightarrow \mathbf{Z}(\mathbb{T}/(\mathbb{U} \setminus \{P\})) \neq \emptyset$.

Algorithm 3: PartialTriangularizeT

Input: A list of form $[T, U, k]$
Output: A list $[T', U', l]$ or $[T', U', l, T'_l, U'_l]$

3.1 $T_k := \{f \in T \mid \mathrm{cls}(f) = k\}$;
3.2 $U_k := \{f \in U \mid \mathrm{cls}(f) = k\}$;
3.3 **if** $T_k = \emptyset$ **then**
3.4 **if** $U_k = \emptyset$ *and* $k > 1$ **then**
3.5 then return $[T, U, k-1]$;
3.6 **else if** $U_k = \emptyset$ *and* $k = 1$ **then**
3.7 then return $[T, U, 0]$;
3.8 **else**
3.9 return $[T, U, \emptyset, U_k, k]$;
3.10 **end**
3.11 **end**
3.12 **while** $\mathrm{nops}(T_k) > 0$ **do**
3.13 $g :=$ *an element of* T_k *with minimal degree in* x_k ;
3.14 **if** $\mathrm{cls}(\mathrm{ini}(g)) > 0$ **then**
3.15 `Branch` $([T\backslash\{g\} \bigcup \{\mathrm{ini}(g), \mathrm{reductum}(g)\}, U, k])$;
3.16 $U := U \bigcup \{\mathrm{ini}(g)\}$;
3.17 $U_k := \{f \in U \mid \mathrm{cls}(f) = k\}$;
3.18 **end**
3.19 **if** $\mathrm{nops}(T_k) = 1$ **then return** $[T, U, k, T_k, U_k]$;
3.20 $f :=$ *an element of* $T_k\backslash\{g\}$;
3.21 $T := T\backslash\{f, g\}$;
3.22 $S := [\mathtt{SubResChain}(f, g)]$;
3.23 # compute the subresultant polynomial remainder #sequence of f and g, denote the subresultant regular #subchain by $H_2, \ldots, H_r$
3.24 **for** $i = 2$ **to** $r - 2$ **do**
3.25 `Branch` $([T \bigcup \{H_i, \mathrm{ini}(H_{i+1}), \ldots, \mathrm{ini}(H_r)\}, U \bigcup \{\mathrm{ini}(H_i)\}, k])$
3.26 **end**
3.27 **if** $\mathrm{cls}(H_r) = k$ **then**
3.28 `Branch` $([T \bigcup \{H_{r-1}, \mathrm{ini}(H_r)\}, U \bigcup \{\mathrm{ini}(H_{r-1})\}, k])$;
3.29 $T := T \bigcup \{H_r\}$;
3.30 $U := U \bigcup \{\mathrm{ini}(H_r)\}$;
3.31 **else**
3.32 $T := T \bigcup \{H_{r-1}, H_r\}$;
3.33 $U := U \bigcup \{\mathrm{ini}(H_{r-1})\}$;
3.34 **end**
3.35 $T_k := \{f \in T \mid \mathrm{cls}(f) = k\}$;
3.36 $U_k := \{f \in U \mid \mathrm{cls}(f) = k\}$;
3.37 **end**

3.2. Computing subresultant PRS

It is an essential step in Wang's algorithm RegSer to compute subresultant PRS. He pointed out that there exists some weakness in Loos's subresultant algorithm in the case when the two input polynomials have the same degree. He implemented a modified version without considering any optimization. In [16], Wang and Xia gave a complete discussion of both cases and presented a more refined version.

The following example shows the differences of the three versions. Let

$$a := a_0x^2 + a_1x + a_2, \qquad b := b_0x^2 + b_1x + b_2.$$

The output (a, b are omitted) of Loos', Wang's, Wang and Xia's method are respectively:

$$\begin{cases} S_1 = -b_0a_1x - b_0a_2 + a_0b_1x + a_0b_2, \\ S_0 = \dfrac{(b_0a_1^2b_2 - a_0b_2a_1b_1 - b_0a_2a_1b_1 + a_2a_0b_1^2 + b_0^2a_2^2 - 2b_0a_2a_0b_2 + a_0^2b_2^2)}{b_0}, \end{cases} \tag{2}$$

$$\begin{cases} S_1 = b_0(b_0a_1x + b_0a_2 - a_0b_1x - a_0b_2), \\ S_0 = b_0(b_0a_1^2b_2 - a_0b_2a_1b_1 - b_0a_2a_1b_1 + a_2a_0b_1^2 + b_0^2a_2^2 - 2b_0a_2a_0b_2 + a_0^2b_2^2), \end{cases} \tag{3}$$

$$\begin{cases} S_1 = b_0a_1x + b_0a_2 - a_0b_1x - a_0b_2, \\ S_0 = b_0a_1^2b_2 - a_0b_2a_1b_1 - b_0a_2a_1b_1 + a_2a_0b_1^2 + b_0^2a_2^2 - 2b_0a_2a_0b_2 + a_0^2b_2^2. \end{cases} \tag{4}$$

Table 1
Timings of SRC and OptSRC.

	Test 1	Test 10	Test 3	Test 2	Ex 2(7)	Test 6	Test 8
SRC	2.656	6.109	62.390	270.295	435.253	>1000	>1000
OptSRC	0.688	4.579	4.250	7.656	22.328	0.203	4.547

Algorithm 4: EliminateU

Input: A list of form $[T, U, k, T_k, U_k]$
Output: A list $[T', U', l]$

4.1 **if** $T_k \neq \emptyset$ **then** $g := T_k[1]$;
4.2 **while** $T_k \neq \emptyset$ *and* $U_k \neq \emptyset$ **do**
4.3 $f := U_k[1]$;
4.4 **if** $\deg(f, x_k) > \deg(g, x_k)$ **then**
4.5 $S := [\texttt{SubResChain}(f, g)]$;
4.6 # compute the subresultant polynomial remainder #sequence of f and g, denote the subresultant regular #subchain by $H_2, \ldots, H_r$
4.7 **else**
4.8 $S := [\texttt{SubResChain}(g, f)]$;
4.9 **end**
4.10 **for** $i = 1$ **to** r **do**
4.11 $h := \text{pquo}(g, H_i, x_k)$;
4.12 Branch $([T \bigcup \{h, \text{ini}(H_{i+1}), \ldots, \text{ini}(H_r)\}, U \bigcup \{\text{ini}(H_i)\}, k])$;
4.13 **end**
4.14 $T_k := T_k \backslash \{g\}$;
4.15 $g := T_k[1]$;
4.16 **end**
4.17 **if** $U_k \neq \emptyset$ *and* $k > 1$ **then**
4.18 **foreach** $P \in U_k$ **do**
4.19 CoefP:=$\{\text{coeffs}(P, x_k)\}$;
4.20 # collect all coefficients of P in x_k
4.21 **if** CoefP contains constants **then** next;
4.22 $f :=$ *an element of* CoefP *with minimal class and minimal terms*;
4.23 Branch $([T \bigcup \{f\}, subs(f = 0, U), k])$;
4.24 U :=$U \bigcup \{f\}$;
4.25 **end**
4.26 **end**
4.27 $[T, U, k - 1]$;

It is easy to see that Loos's version of subresultant algorithm does not always obtain correct result. Wang's version may contain some redundant factors. Wang and Xia's version obtains the most satisfactory result. We implemented Wang and Xia's version with Ducos' optimization strategy [10] in Maple. More work on the computation of subresultants is in progress.

3.3. Experiments

We make several experiments to show how much Ducos's optimization strategy can improve the subresultant algorithm. Table 1 gives the timings of problems solved by OptSRC and SRC which denote respectively Wang and Xia's subresultant algorithm with or without optimization. These benchmarks are from [10] and [17]. From this table one can see that, the optimization strategy of Ducos is quite powerful.

Table 2 illustrates the timings of RegSer, NewRegSer and RecurWeakRegSer for decomposing polynomial sets into (weak) regular systems. NewRegSer denotes the new variant of RegSer with optimized subresultant algorithm. In this table, Num denotes the number of the components of the output. These benchmarks are from [18,19] and references therein. All the timings (in second) are run on Intel Pentium 4 (3.20 GHz CPU, 512 MB memory) and Maple 11.

4. Conclusion

In this paper, we generalized the notion of regular systems to that of weak regular systems and studied various properties of weak regular systems. We presented a recursive algorithm RecurWeakRegSer to compute a weak regular series of a given polynomial set or system. The implementation of this algorithm in Maple turns out to be more efficient than RegSer. Moreover, RecurWeakRegSer can handle problems which cannot be solved by RegSer. An optimized version of subresultant

Table 2
Timings of RegSer, NewRegSer and RecurWeakRegSer.

Name	RegSer		NewRegSer		RecurWeakRegSer	
	Time (s)	Num	Time (s)	Num	Time (s)	Num
Montes-S12	0.344	7	0.344	7	0.280	7
FourCircles	0.781	8	0.547	3	0.515	3
Montes-S16	1.875	15	1.125	15	0.766	15
Caprasse	2.625	4	1.406	4	0.923	4
Gonnet-83	4.218	18	1.953	7	1.062	6
Wu-87	7.265	6	6.266	6	5.031	6
Gerdt-91b	8.610	7	3.313	6	2.937	5
Rose-1	10.250	1	9.702	1	9.312	1
Example 17	11.938	2	5.375	2	5.953	2
Example 43	13.265	1	10.249	1	5.984	1
Montes-S11	27.217	25	15.719	24	14.125	19
Gonnet-83-1	31.562	127	10.531	54	3.907	52
Czapor-87-1	>1000		29.296	1	44.047	1
Montes-S14	>1000		>1000		>1000	

algorithm was implemented in Maple. Experimentation results indicate that it does improve the efficiency of RegSer when this version is called, sometimes it can save half of the time.

Acknowledgements

The author would like to thank his supervisor Prof. Dongming Wang who introduced him to the theory of regular decomposition. The author would also like to thank the two anonymous reviewers for their useful comments.

References

[1] D. Wang, Computing triangular systems and regular systems, J. Symbolic Comput. 30 (2) (2000) 221–236.
[2] M. Kalkbrener, Three contributions to elimination theory, Ph.D. thesis, Johannes Kepler University, Linz, 1991.
[3] L. Yang, J. Zhang, Searching dependency between algebraic equations: An algorithm applied to automated reasoning, in: J. Johnson, S. McKee, A. Vella (Eds.), Artificial Intelligence in Mathematics, Oxford University Press, Oxford, 1994, pp. 147–156.
[4] D. Lazard, Solving zero-dimensional algebraic systems, J. Symbolic Comput. 15 (1992) 117–132.
[5] M. Moreno Maza, On triangular decomposition of algebraic varieties, Technial Report TR 4/99, NAF Ltd, Oxford, UK, 1999.
[6] E. Hubert, Notes on triangular sets and triangulation-decomposition algorithms I: Polynomial systems, in: F. Winkler, U. Langer (Eds.), Symbolic and Numerical Scientific Computing 2001, in: Lecture Notes in Computer Science, vol. 2630, Springer Verlag, Heidelberg, 2003.
[7] D. Wang, Decomposing polynomial systems into simple systems, J. Symbolic Comput. 25 (3) (1998) 295–314.
[8] B. Xia, X. Hou, A complete algorithm for counting real solutions of polynomial systems of equations and inequalities, Comput. Math. Appl. 44 (2002) 633–642.
[9] J. Zhang, L. Yang, X. Hou, A note on Wu Wen-Tsun's nondegenerate condition, Chinese Sci. Bull. 38 (1) (1993) 86–87.
[10] L. Ducos, Optimization of the subresultant algorithm, J. Pure Appl. Algebra 145 (2000) 149–163.
[11] D. Wang, Elimination Methods with Applications, Science Press, Beijing, 2002 (in Chinese).
[12] W. Sit, Computations on quasi-algebraic sets, in: L. Liska (Ed.), Electronic Proceedings of IMACS ACA' 98, 1998.
[13] D. Bousiane, A. Rody, H. Maârouf, Unmixed-dimensional decomposition of a finitely generated perfect differential ideal, J. Symbolic Comput. 31 (2001) 631–649.
[14] W. Sit, The Ritt–Kolchin theory for differential polynomials, in: L. Guo, P. Cassidy, W. Keigher, W. Sit (Eds.), Differential Algebra and Related Topics: Proceedings of International Workshop, 2002, pp. 1–70.
[15] C. Chen, M. Moreno Maza, W. Pan, Y. Xie, On the verification of polynomial system solvers, in: T. Ida, Q. Jiang, D. Wang (Eds.), Proceedings of the Fifth Asian Workshop on Foundations of Software, 2007, pp. 116–144.
[16] D. Wang, B. Xia, Computer Algebra, Tsinghua University Press, Beijing, 2004 (in Chinese).
[17] X. Hou, D. Wang, Subresultants with the Bezout matrix, in: X. Gao, D. Wang, (Eds.), Computer Mathematics: Proceedings of the Fourth Asian Symposium (ASCM 2000) pp. 19– 28.
[18] M. Manubens, A. Montes, Improving the DISPGB algorithm using the discriminant ideal, J. Symbolic Comput. 41 (2006) 1245–1263.
[19] D. Wang, Elimination Practice, Imperial College Press, London, 2004.